Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2018

The Brew Crew
Doom Bot

Peter Dang
Matthew Draft
Eddie Molina
Van Ta
Marco Paramo
Robert Villazana

Revision History

Revision	Date	Author(s)	Description
0.1	10.01.2015	VT	document creation
0.2	08.06.2018	MD	Corrected Front Page, added brain layer, updated images
0.3	08.08.2018	PD	Added System Overview
0.4	08.09.2018	RV	Added Error Detection subsystem
0.5	08.09.2018	EM	Added Motor control subsystem
0.6	08.09.2018	MP	Added Interface subsystem

Contents

List of Figures

List of Tables

1 Introduction

The DOOM hopper will provide any home or commercial brewing system with some automated boil management. The goal is to read a recipe and according to the recipe it will add and remove hops. The recipe will be documented in a '*.bs' file, and this system will have 4 layers: brain, interface, motor control and miscellaneous system. The brain will serve as a central hub for the entire system. The motor control will handle the physical aspects like transporting the hops to the kettle through a pully system and motors. The interface acts as a menu for the user so they can select the appropriate recipe and pulleys for their brew. Miscellaneous system is components that require no input or output but required for the structural integrity of the system.

2 System Overview

The Brain Layer will serve as the top-level layer as to which it controls the machine. The way the Brain Layer interacts with the user is through the Interface layer where it asks them which recipe to choose upon. Once chosen, the information will go back to the Brain Layer and send signals to the Motor Control Layer and proceed with dumping hops at the correct time period. Once an objective is completed in the Motor Control Layer, it will send a completion message to the Brain Layer and proceed with the next step in the process until completion. A Misc. Systems Layer was also added where it provides a list of necessary products outside of the machine in order for the machine to operate.

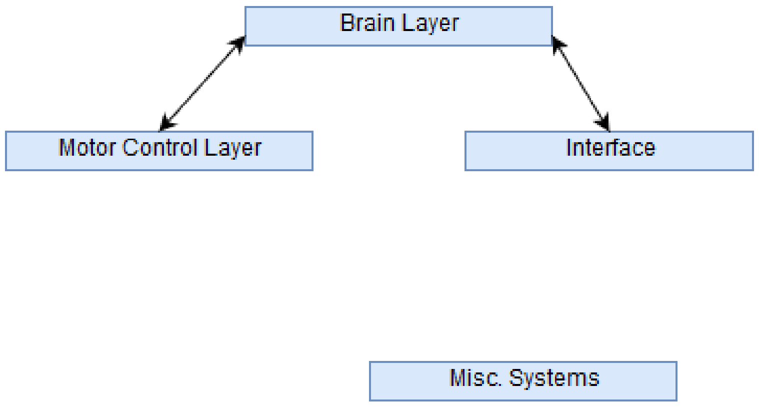

Figure 1: A simple architectural layer diagram

2.1 Brain Layer

The Brain Layer operates as the systems command functionality in which it sends messages through the system for it to complete an objective. Two subsystems that are used for this layer is the Raspberry Pi and a Timer. Once turned on, the Brain Layer should send recipes to the Interface to choose upon and waits for a user response. Once chosen, a Timer will start and the Raspberry Pi will follow each objective in the recipe and send signals to the Motor Control.

2.2 Interface Layer

Prior to the operation of the Interface Layer, recipes should already be installed in the Brain Layer. The Interface Layer will have two subsystems, the Android App and the Webpage. In the beginning of operation, the Brain Layer will send a list of recipes to the Android App. When the Android App is running, the user should be able to select upon a list of recipes obtained from the Brain Layer. Once chosen, the interface will send the recipe selection to the Brain Layer to work with. Once the timer is started, a webpage from the interface should display the cumulative time of the entire process.

2.3 Motor Control Layer

The Motor Control Layers objective is to physically do the objective given from the Brain Layer. The Motor Control Layer has four subsystems, Error Detection, Motors, Pulley, and Clips. From the Raspberry Pi, a specific motor set should be chosen to work with for the specific objective. The motor should run with a pulley transferring a string with hops down into the kettle. Once the objective is completed, the motors will go in reverse and take the hops away from the kettle. When that is completed, the motor control will check if there was any error detection. If not, then the layer will send a message to the Raspberry Pi saying that the objective was completed and is available for the next one.

2.4 Misc. Systems Layer

The Misc. Systems Layer are objects required for the project for the machine to work around with. There are three subsystems, the Kettle, Drip tray, and stand. The Kettle will be where the brewing takes place. The Drip tray is for finished hops being taken out for cleanliness. The stand is for display purposes of the machine.

3 Subsystem Definitions & Data Flow

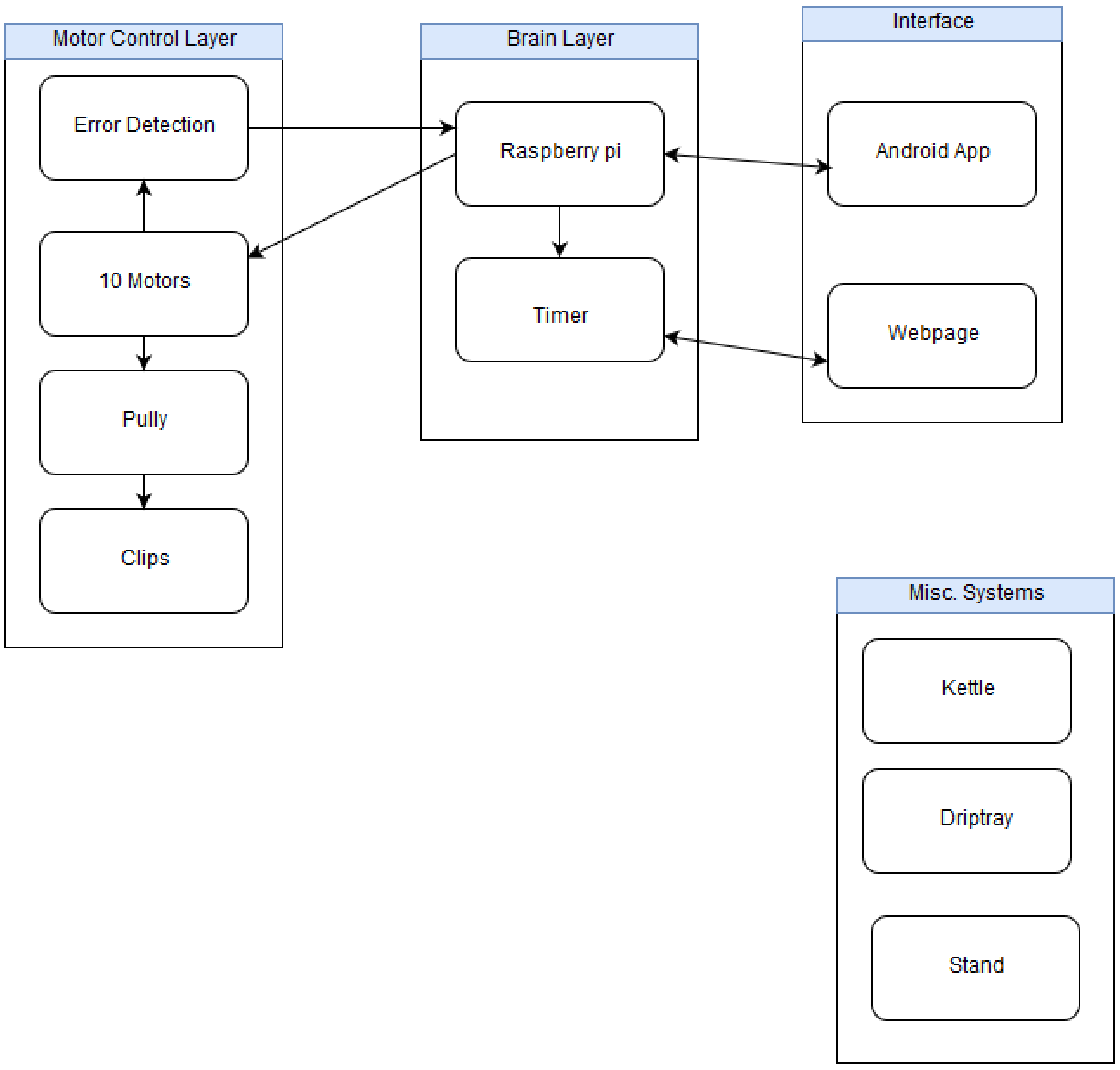

Figure 2: A simple data flow diagram

4 Brain Layer Subsystems

The Brain layer serves as a central hub for data exchange in the device. It should be controlling when hops are added to the system and for how long they steep. It will also be responsible for taking commands from the user.

4.1 Raspberry Pi

This serves as the main control device for the unit. It will host the webpage and run all calculations for the device. It will also activate the motors.

4.1.1 Assumptions

The assumption we are making is that the raspberry pi can securely and reliably host webpages and make calculations quickly enough for this system.

4.1.2 Responsibilities

The Raspeberry pi will be responsible for all server side calculations as well as the main functionality of the device out side of the user interface.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#00	Controls the timer	Timer Control	output 1

4.2 Timer

This subsystem is soly responsible for keeping the time. Since this is an incredibly important function for brewing we must make sure we do it correctly.

4.2.1 Assumptions

We assume it is possible to keep accurate time.

4.2.2 Responsibilities

This is responsible for the timed events that must happen at certain times.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Controls the timer	Signal Reception	input 1

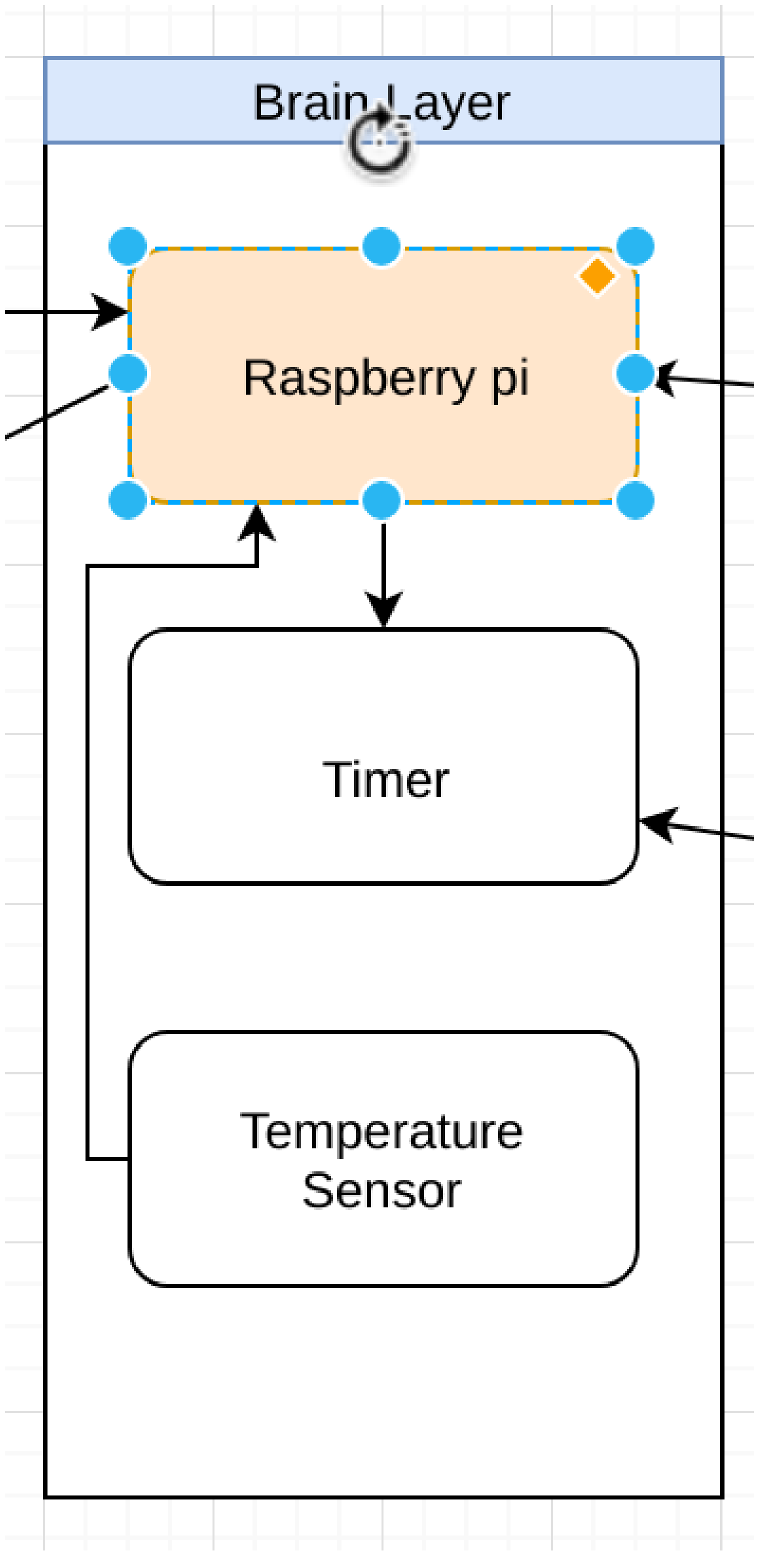

Figure 3: Raspberry Pi Subsystem

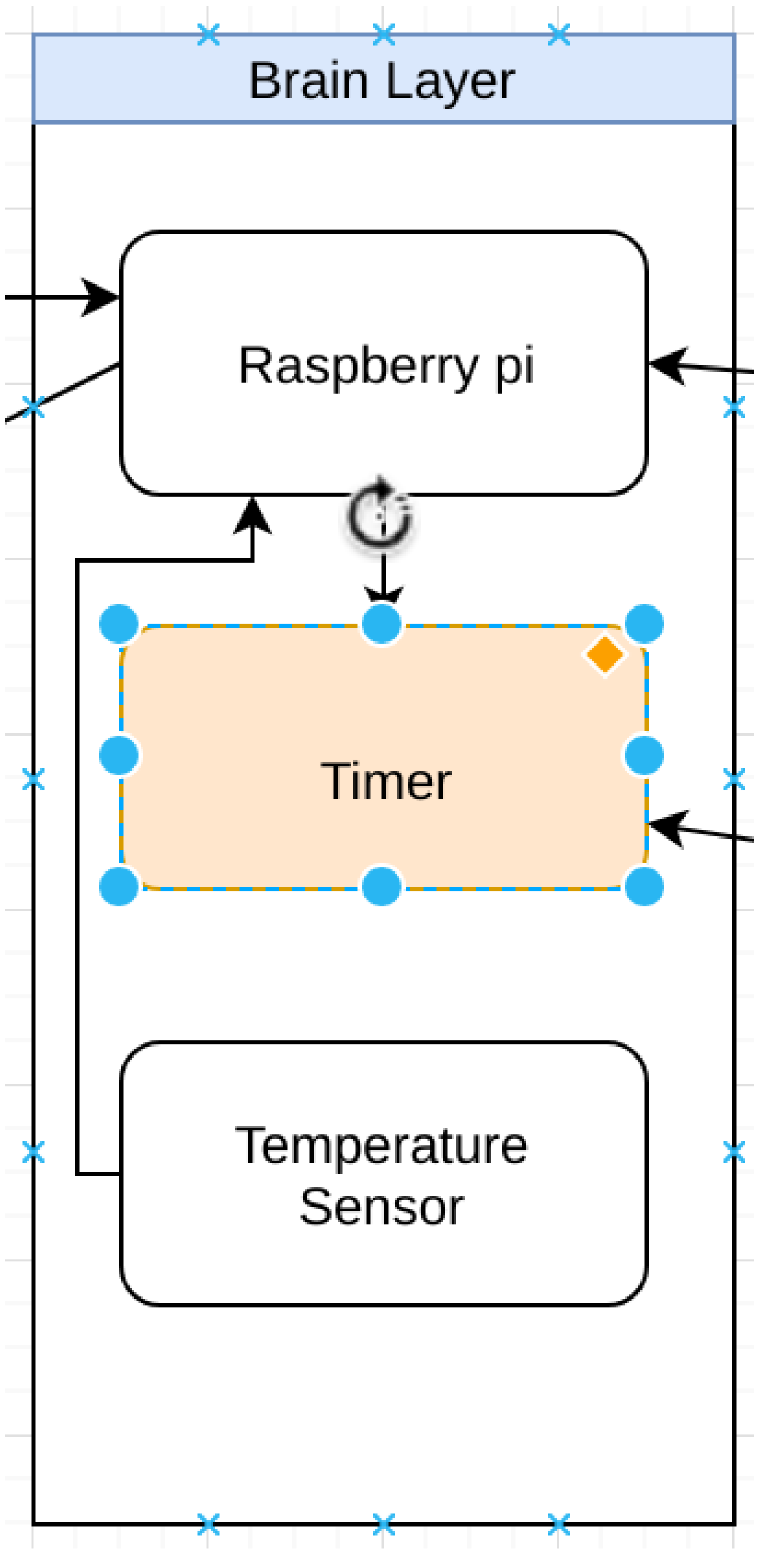

Figure 4: Timer

5 Motor Control Subsystems

The Motor Control layer is the layer where the physical action happens. This layer will have two sub-systems, the first system is Motors. The Motors sub-system will have approximately ten motors that will control, what The Brew Crew likes to call the DOOM Hopper, using a Raspberry Pi. The DOOM Hopper will be in charge of dropping the brew hops into the kettle at a specific time. The second subsystem is Error Detection. The Error Detection subsystem will be keeping track of all the motors. In other words, the subsystem will check if all the motors are moving at the programmed rate and be positioned at the correct position.

5.1 Error Detection

The error detection subsystem will be powered the Brain layer, specifically, by the raspberry pi. Error detection will always be checking on the motors of the system, it will check the RPMs of the motors and fix the RPM if too slow or too fast. Error detection will also keep track of the position of the motor; with proper position of the motors the hops can be dropped correctly into the boiling kettle.

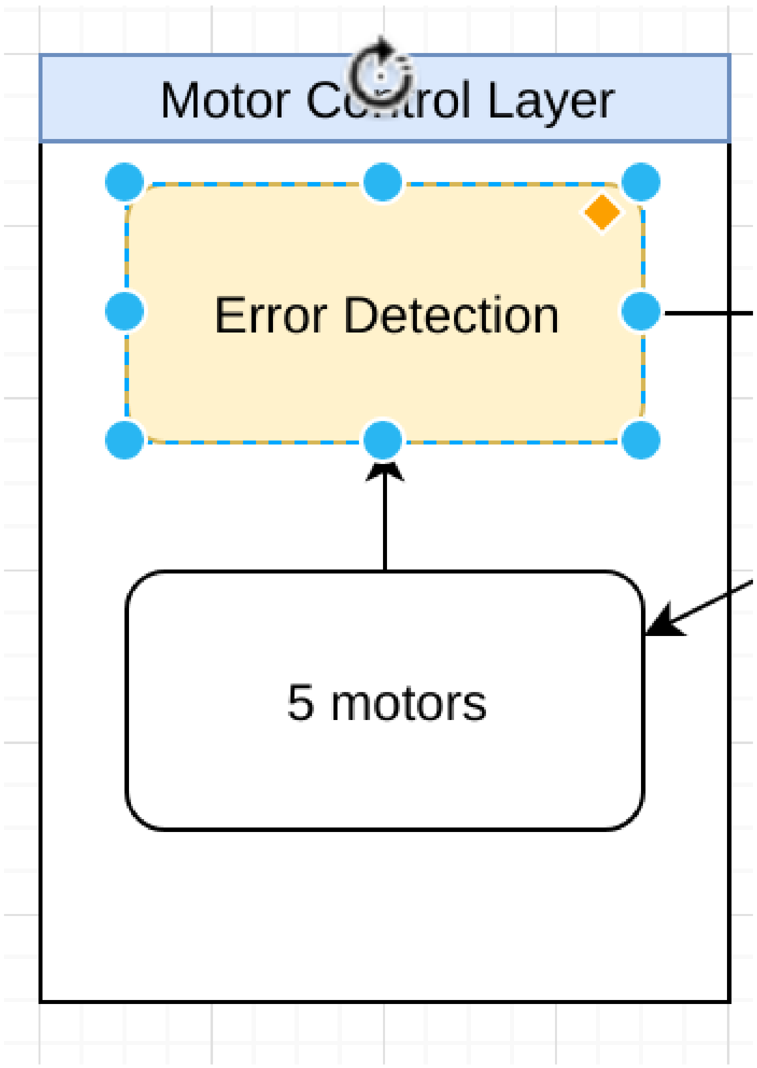

Figure 5: Error Detection Diagram

5.1.1 Assumptions

We assume error detection will be controlled by the Raspberry Pi.

5.1.2 Responsibilities

Error detection is responsible for keeping all motors at the right RPM and in the right position.

5.1.3 Subsystem Interfaces

The subsystem interface is the raspberry pi from the Brain Layer.

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Detects errors	Raspberry Pi	Sends positive or negative confirmation to Raspberry PI

5.2 Motors

The motors are the moving forces of the hop system. The objective of the motors is to move the hop dispensers into positon for displacing the hops and specialty grains into the stainless steel kettle for boiling. The user will select a recipe from the interface layer, which consist of a web page or an android application. Upon selection, only the hops in the dispensers needed for the recipe will be dumped into the kettle. This is important for the motor control since the user selection will determine how long the motors should run in order to position the right hop dispensers in front of the kettle.

5.2.1 Assumptions

The assumptions for the motor subsystem is that there will be approximately 10 motors controlling the hop containers. Also, the motors will be controlled by the brain system, which consist of a raspberry pi.

5.2.2 Responsibilities

The motor subsystem is responsible for turning each hop container at a certain rpm. It will allow for the hops to be dumped at the right position.

5.2.3 Subsystem Interfaces

The subsystem's interface is the raspberry pi and the timer, which come from the Brain Layer.

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Motor Control	Raspberry Pi	Hop Dispenser Position

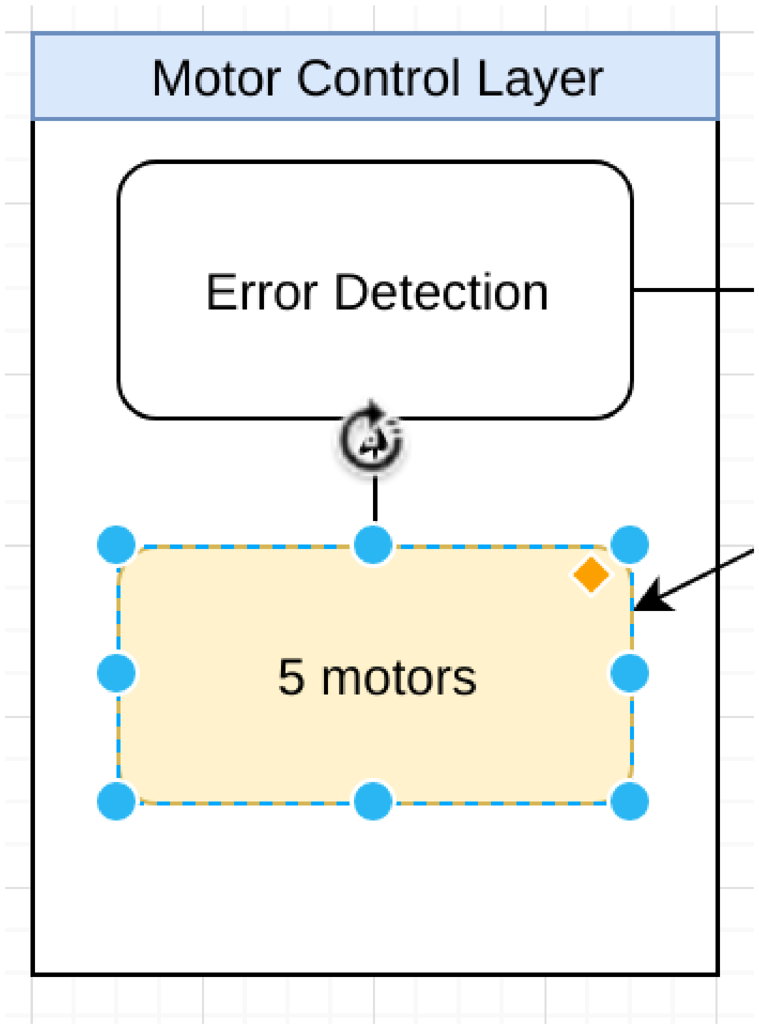

Figure 6: Motor Control Diagram

6 Interface Subsystem

The interface layer will serve as the software structure of the brewing system. It will act similar to a menu for the user to choose their recipe through the android app and will connect to a server from the raspberry pi or a webpage, possibly both.

6.1 Android App

The Android App will communicate with the raspberry pi to basically provide an interface for the user to utilize the hop machine.

6.1.1 Assumptions

The android app will provide recipes to choose from and communicate with the brain.

6.1.2 Responsibilities

The android app should have a menu like interface for the user to be able to easily understand and use. The app should be in sync with the brain at all times.

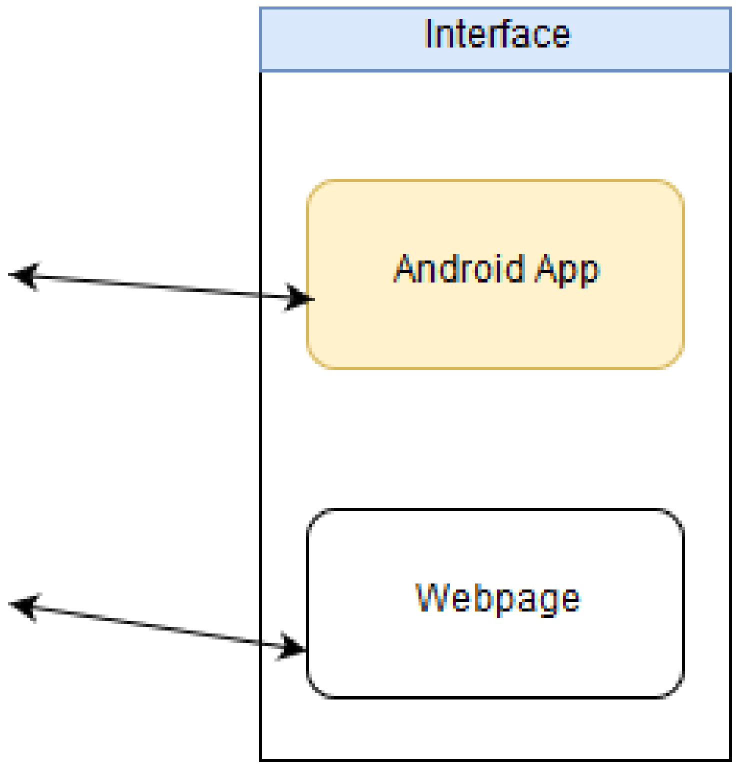

Figure 7: Example subsystem description diagram

6.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Start and Stop	User Input	Signal to Pi
#02	Send recipe to the webpage	User Input File	Parsed recipe class
$03	Receive updates from webpage	Status class	Updated status class

6.2 Webpage

It will act as the same thing as the android app and as a possible substitute. It will allow users to access the app and alternately anything connected to the internet.

6.2.1 Assumptions

It basically does the same thing as the android app and will allow users to use either app other devices to access the brewing system. Will give the user more direct control over the entire system.

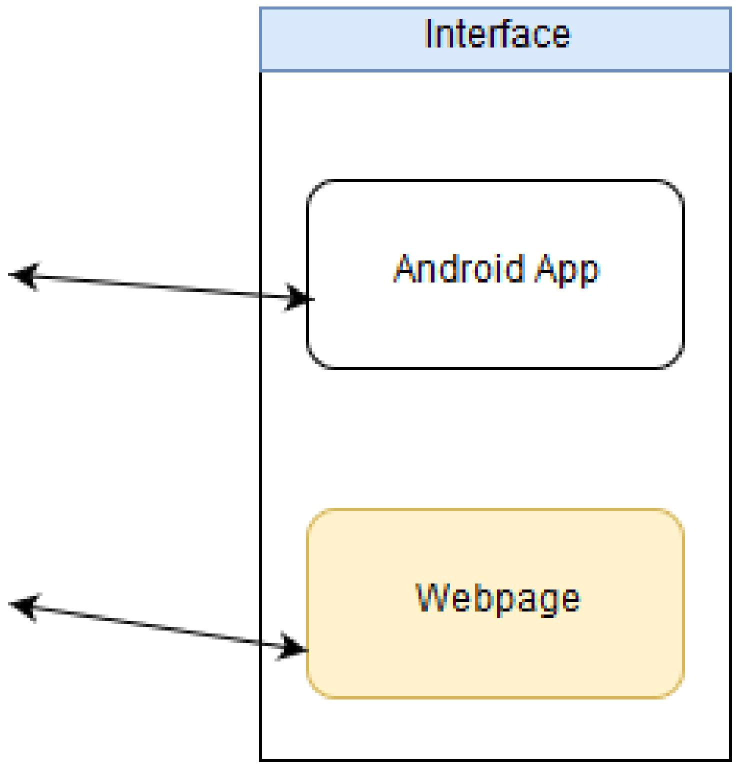

Figure 8: Example subsystem description diagram

6.2.2 Responsibilities

The webpage will be a single page ran on the Raspberry Pi. This subsystem will be able to display any important information and manually control the system. The page will be split up into sections to help the user understand the system in its current state.

6.2.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Send signal to dispense hops	User input	Send a command
#02	Display updates	Data class from Pi	N/A
#03	Send new recipes to database	Recipe File	Parsed File